جامعة محمد بوضياف - المسيلة
Université Mohamed Boudiaf - M'sila

# Artificial Learning Models

## Lecture 1 : Machine Learning Overview

By : Prof. Lamri SAYAD

2024

# Agenda

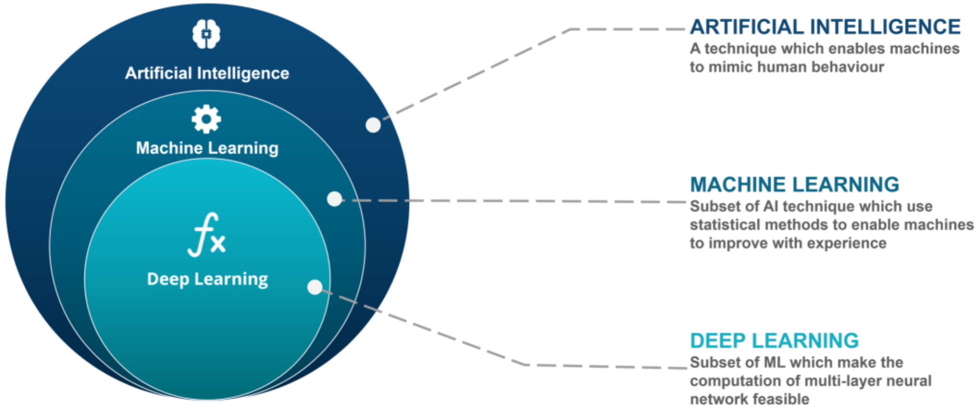
- Introduction
- Terminologies and Definitions
- History
- Application Areas
- Some standard learning tasks
- Types of Machine Learning
  - Supervised Machine Learning
  - Unsupervised Machine Learning
  - Reinforcement Machine Learning
- When to Use Machine Learning?

# Introduction

# Introduction


*"**AI** is the science and engineering of making intelligent machines." — John McCarthy*

*"**Machine learning** is the field of study that gives computers the ability to learn without being explicitly programmed." — Arthur Samuel*



# Terminologies & definitions

# Terminologies and Definitions

- Let’s consider a dataset of Emails

| Sender | Destination | Date | …. | Text | Type |
|---|---|---|---|---|---|
| abc@aaa.dz | defg@bbb.dz | 24/09/2023 | …. | Bla bla bla | Spam |
| … | | | | | Email |
| … | | | | | |
| | | | | | Spam |

# What is machine learning?

- Algorithms that can learn from observational data, and can make predictions based on it.

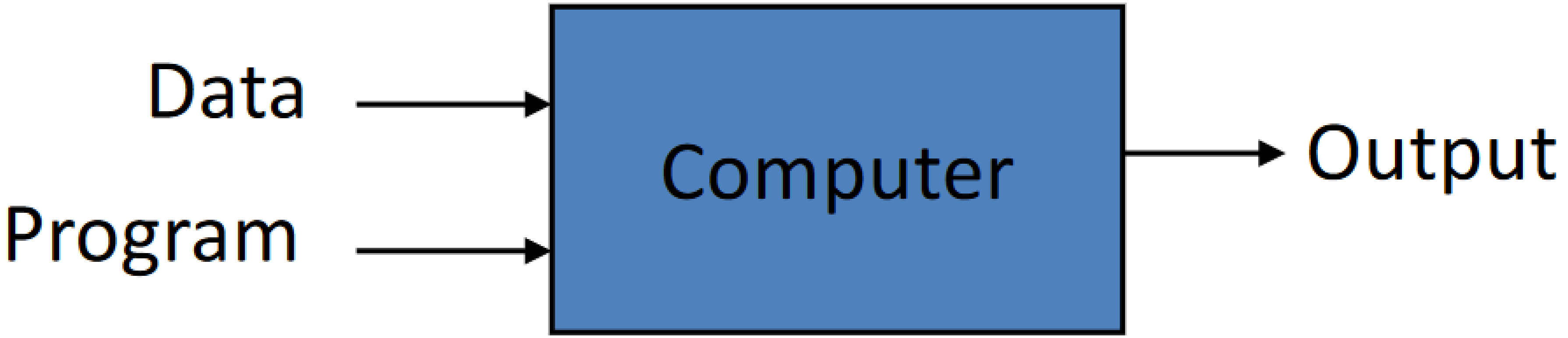
that’s pretty much what your own brain does too.

# What is machine learning?

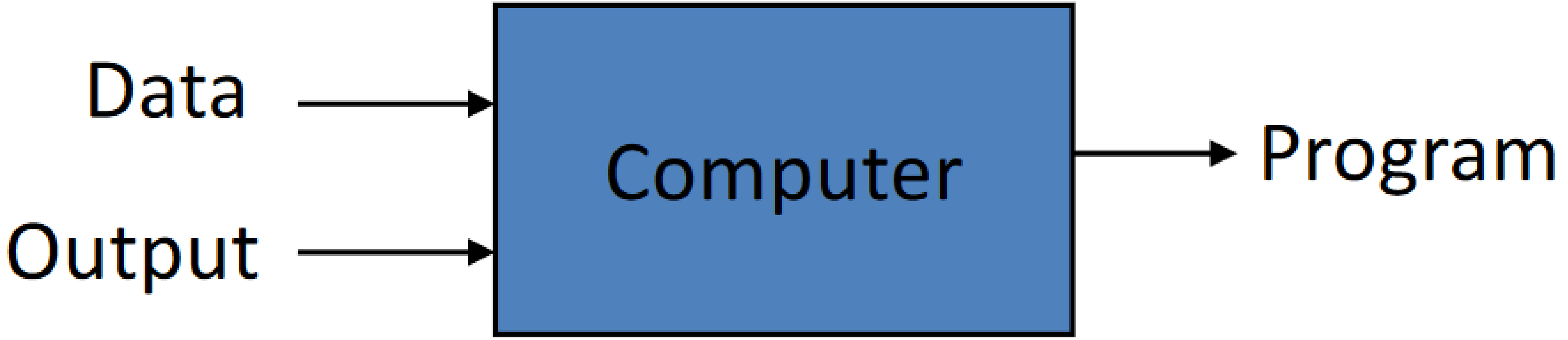
- **Machine learning** is a subfield of computer science that is concerned with building algorithms that, to be useful, rely on a collection of examples of some phenomenon. These examples can come from nature, be handcrafted by humans, or generated by another algorithm.

- **Machine learning** can also be defined as the process of solving a practical problem by,
    1. collecting a dataset, and
    2. algorithmically training a statistical model based on that dataset

## Traditional Programming

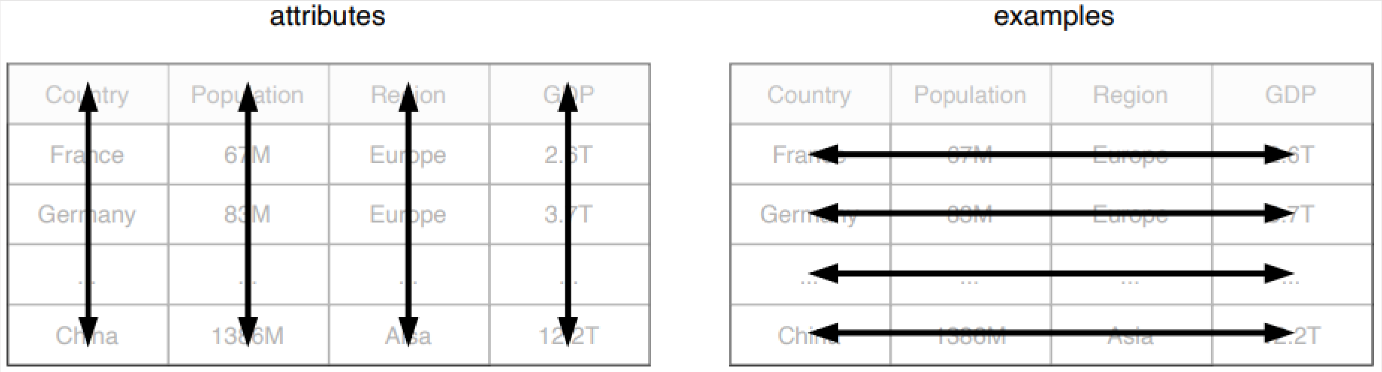



## Machine Learning



# Terminologies and Definitions

- **Examples**: Items or instances of data used for learning or evaluation.
- 
- **Features**: The set of attributes associated to an example.




- **Labels**: Values or categories assigned to examples.
  - *In classification problems*: examples are assigned specific categories, for instance, the spam and non-spam categories in our binary classification problem.
  - *In regression*: items are assigned real-valued labels

# Terminologies and Definitions

- **Hyperparameters**: Free parameters that are not determined by the learning algorithm, but rather specified as inputs to the learning algorithm.

- **Training sample**: Examples used to train a learning algorithm.

- **Validation sample**: Examples used to tune the parameters of a learning algorithm when working with labeled data. The validation sample is used to select appropriate values for the learning algorithm's free parameters (hyperparameters).

# Terminologies and Definitions

- **Test sample**: Examples used to evaluate the performance of a learning algorithm.
  - is separated from the training and validation data
  - and is not made available in the learning stage.

- **Loss function (cost function)** : is a method of evaluating how well your machine learning algorithm models your featured dataset. Denoting the set of all labels as Y and the set of possible predictions as Y0, a loss function $L$ is a mapping $L : y \times y_0 \rightarrow R^+$. In most cases, $y = y_0$ and the loss function is bounded, but these conditions do not always hold. Common examples of loss functions include
  - the zero-one (or misclassification) loss defined over {-1; +1} x {-1; +1} by L(y; y0) = 1 if y0≠y
  - and the squared loss defined by $L(y, y_0) = (y_0 - y)^2$.

# Terminologies and Definitions

- A **machine learning pipeline** is a sequence of operations on the dataset that goes from its initial state to the model.

- A **pipeline** can include, among others, such stages as data partitioning, missing data imputation, feature extraction, data augmentation, class imbalance reduction, dimensionality reduction, and model training

# History of Machine Learning

# History
## The early days

- 1943
  - first mathematical model of neural networks
  - presented in the scientific paper "A logical calculus of the ideas immanent in nervous activity" by **Walter Pitts and Warren McCulloch**,
- 1949
  - the book The Organization of Behavior by Donald Hebb is published.
  - The book had theories on how behavior relates to neural networks and brain activity and would go on to become one of the monumental pillars of machine learning development.
- In 1950
  - Alan Turing created the Turing Test to determine if a computer has real intelligence.
  - presented the principle in his paper Computing Machinery and Intelligence while working at the University of Manchester.
  - It opens with the words: "*I propose to consider the question, 'Can machines think?'* "

# History
## Playing games

- 1952 : The first ever computer learning program was written by Arthur Samuel. The program was the game of checkers.
- 1957 Frank Rosenblatt designed the first neural network for computers - the perceptron –
- 1967 the “nearest neighbor” algorithm was written
- 1979 students at Stanford University invent the ‘Stanford Cart’ which could navigate obstacles in a room on its own
- 1981, Gerald Dejong introduced the concept of Explanation Based Learning (EBL), where a computer analyses training data and creates a general rule it can follow by discarding unimportant data.

# History

## Big steps forward

- In the 1990s, work on machine learning shifted from a knowledge-driven approach to a data-driven approach.
- in 1997, IBM's Deep Blue shocked the world by beating the world champion at chess.
- The term "deep learning" was coined in 2006 by Geoffrey Hinton to explain new algorithms that let computers "see" and distinguish objects and text in images and videos.
- in 2010 Microsoft revealed their Kinect technology could track 20 human features at a rate of 30 times per second, allowing people to interact with the computer via movements and gestures.
- The follow year IBM's Watson beat its human competitors at Jeopardy.
- Google Brain was developed in 2011
- In 2014, Facebook developed DeepFace, a software algorithm that is able to recognize or verify individuals on photos to the same level as humans can.

# History

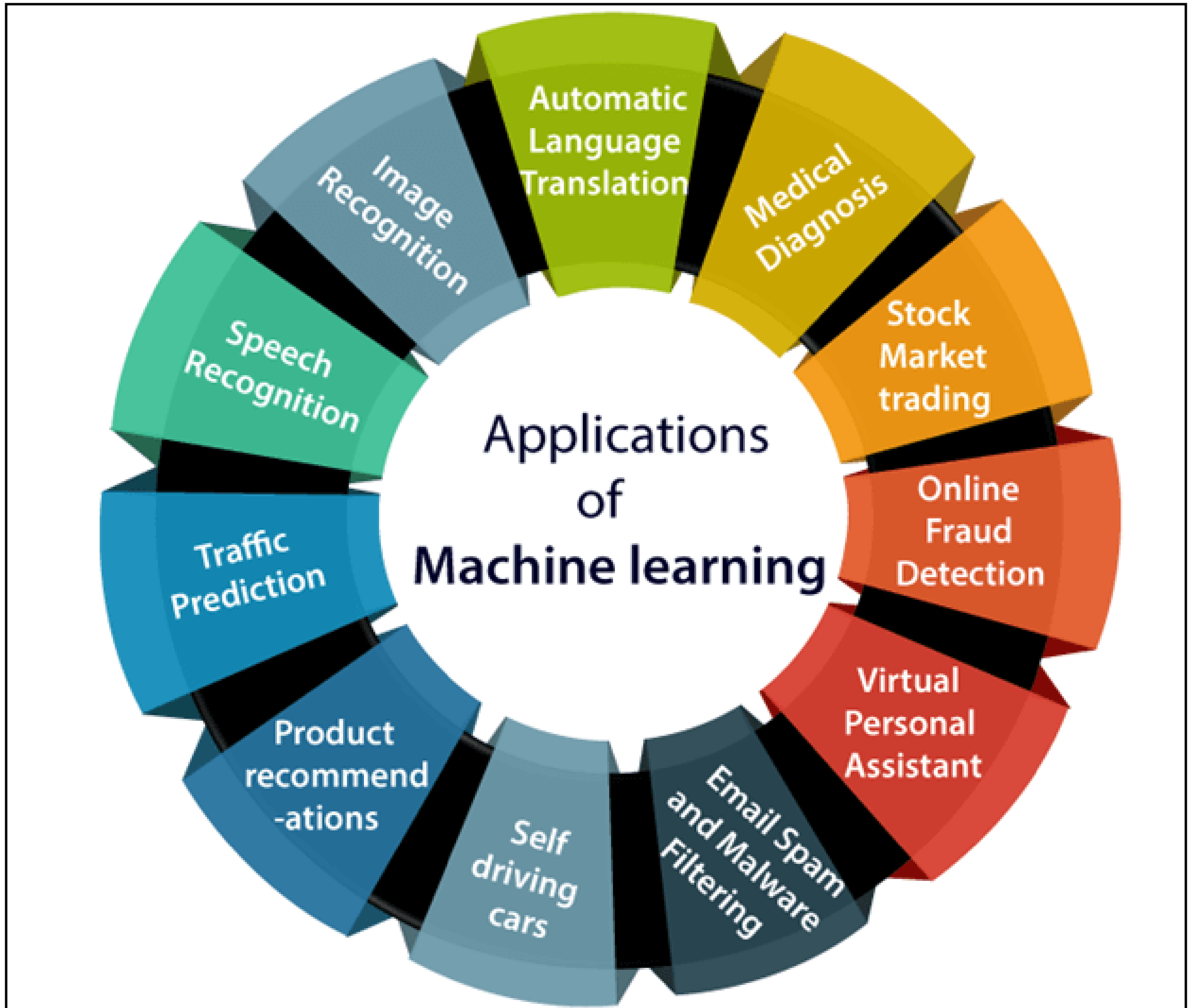
## 2015 – Present days

- Amazon launched its own machine learning platform in 2015. Microsoft also created the Distributed Machine Learning Toolkit
- In 2016 Google's artificial intelligence algorithm beat a professional player at the Chinese board game Go
- Waymo started testing autonomous cars in the US in 2017 with backup drivers only at the back of the car.
- In 2020, while the rest of the world was in the grips of the pandemic, open AI announced a ground-breaking natural language processing algorithm GPT-3 with a remarkable ability to generate human-like text when given a prompt

# The future of machine learning

- **Improvements in unsupervised learning algorithms**
  - In the future, we'll see more effort dedicated to improving unsupervised machine learning algorithms to help to make predictions from unlabeled data sets.
- **The rise of quantum computing**
  - Quantum computers lead to faster processing of data, enhancing the algorithm's ability to analyze and draw meaningful insights from data sets.
- **Focus on cognitive services**
  - Features such as visual recognition, speech detection, and speech understanding will be easier to implement. We're going to see more intelligent applications using cognitive services appear on the market.

# Application areas



# The ML Approach

Data Collection

Samples

Model Selection

Probability distribution to model process

Parameter Estimation

Values/distributions

Inference

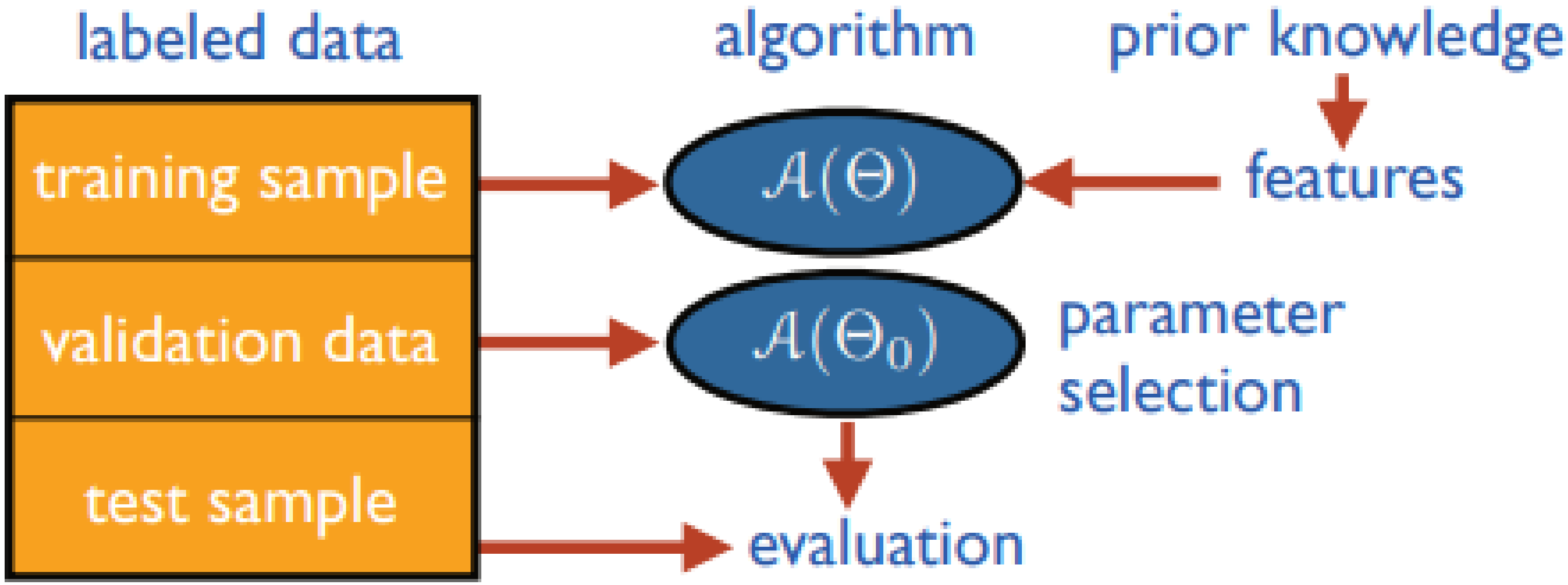
Find responses to queries

Generalization

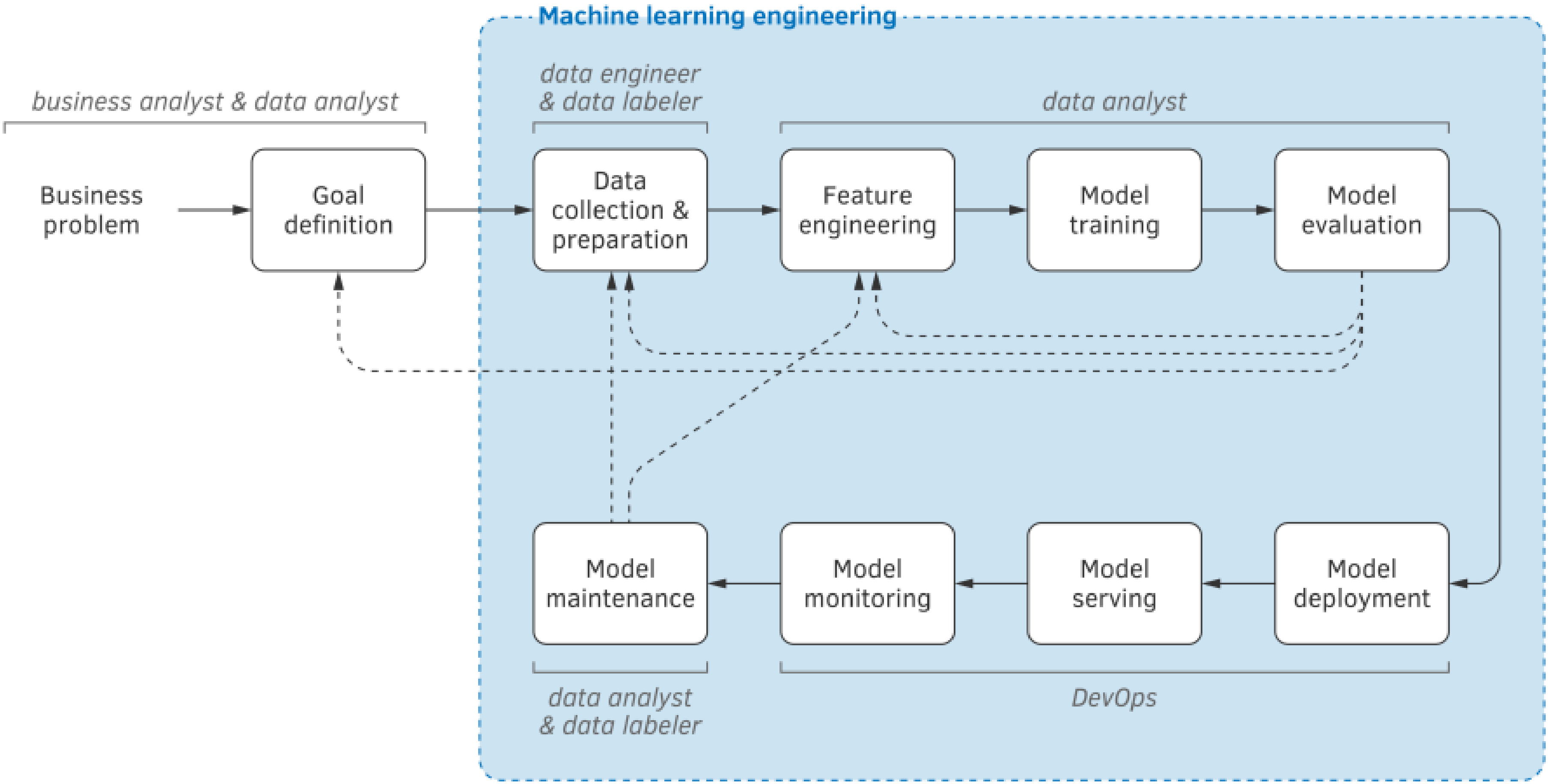
(Training)

Decision
(Inference
OR
Testing)

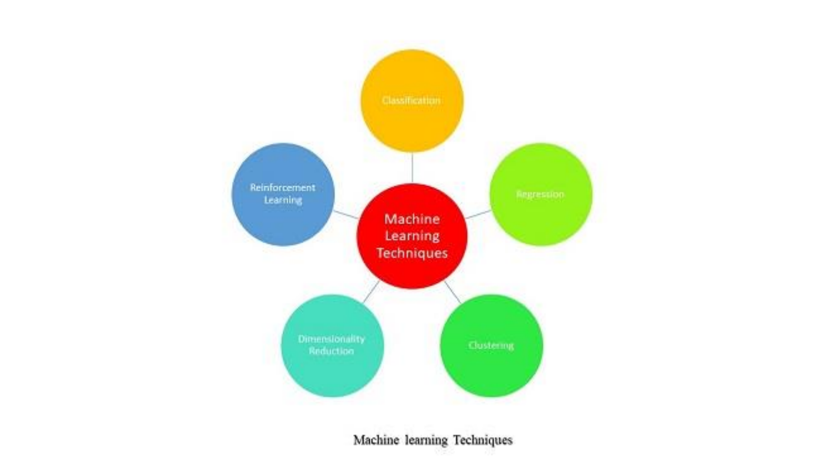
# Learning stages



# ML project life cycle



# Some learning tasks

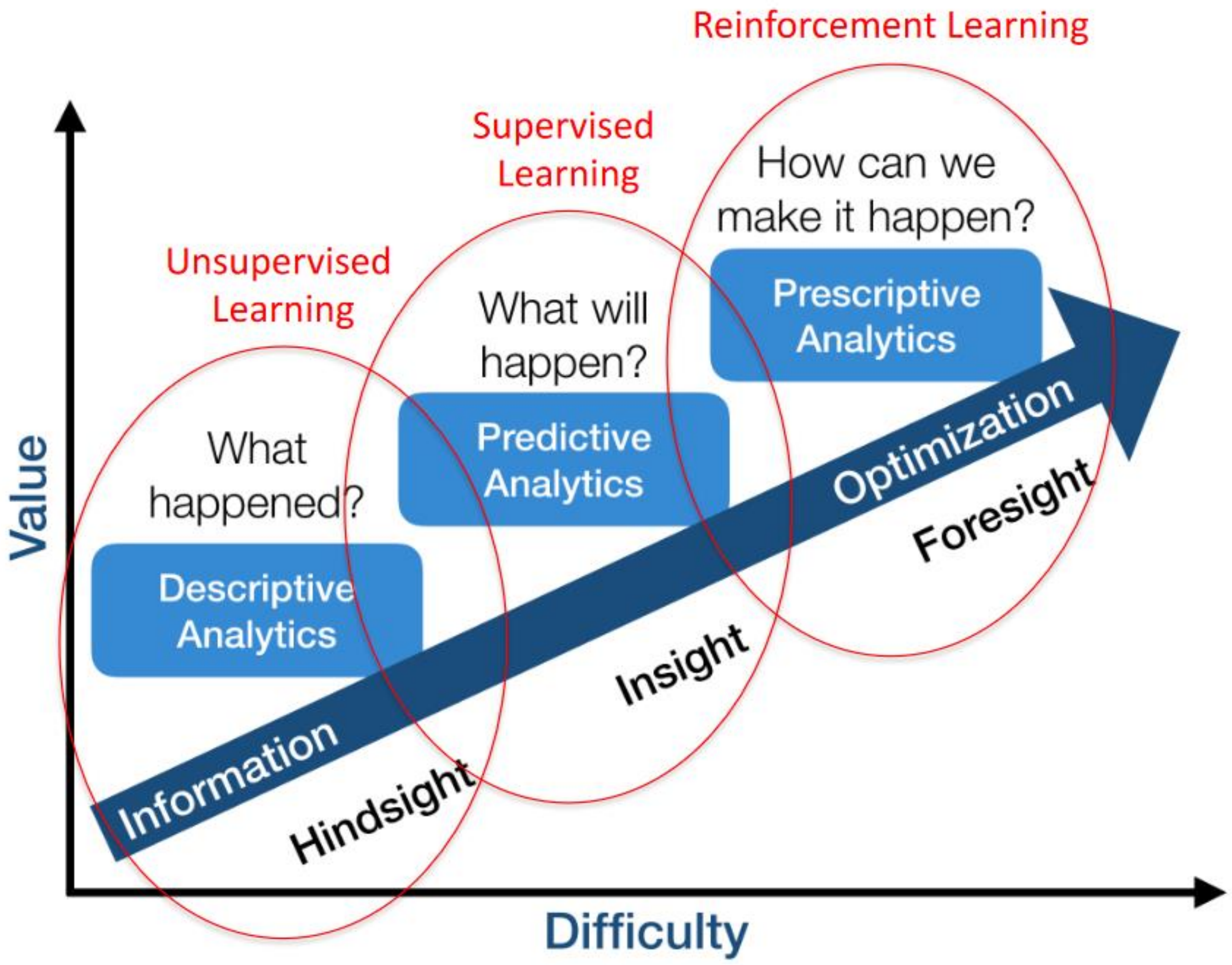



- Classification
  - Binary
  - Multiclass
- Regression
- Clustering
- Anomaly detection
- Dimensionality reduction or manifold learning
- Ranking : this is the problem of learning to order items according to some criterion.
- Recommandation
- Forcasting

# Types of Machine Learning

- Supervised Learning: correct output known for each training example
  - **Classification**
  - **Regression**
  - **Forecasting**
- Unsupervised Learning: Create an internal representation of the input, capturing regularities/structure in data
  - **Clustering**
  - **Dimension reduction**
- Reinforcement Learning : Learn action to maximize reward

# Types of learning



# Supervised learning

- Most widely used methods of ML, e.g.,
  - Spam classification of email
  - Face recognizers over images
  - Medical diagnosis systems
- **Classification**: The output variable to be predicted is *categorical* in nature, e.g.classifying incoming emails as spam or ham, Yes or No, True or False, 0 or 1.
- **Regression**: The output variable to be predicted is *continuous* in nature, e.g. scores of a student, gold prices, etc.

# Supervised learning

- The data the algorithm "learns" from comes with the "correct" answers
- A collection of labeled examples $D = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(N)}, y^{(N)})\}$
  - Each element xi among N is called a feature vector.
  - In computer science, a vector is a one-dimensional array. The length of that sequence of values, D, is called the vector's dimensionality.
- The model created is then used to predict the answer for new, unknown values.
- Example: You can train a model for predicting car prices based on car attributes using historical sales data. That model can then predict the optimal price for new cars that haven't been sold before.

# Unsupervised learning

- the dataset is a collection of unlabeled examples $D = \{x^{(1)}, x^{(2)}, \ldots, x^{(N)}\}$
  - Again, x is a feature vector,
- the goal of an unsupervised learning algorithm is to create a model that takes a feature vector x as input and either transforms it into another vector or into a value that can be used to solve a practical problem.

# Unsupervised learning

- The model is not given any “answers” to learn from; it must make sense of the data just given the observations themselves.

- Example: group (cluster) some objects together into 2 different sets. But I don’t tell you what the “right” set is for any object ahead of time.

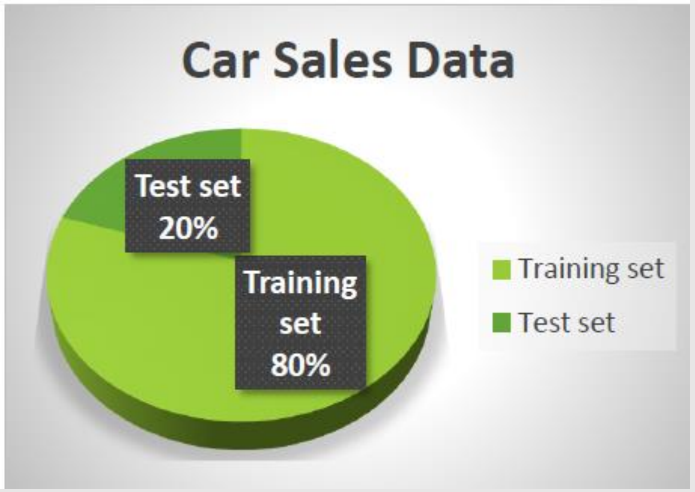
# Reinforcement learning

- is a subfield of machine learning
- where the machine (called an agent) “lives’ ’ in an environment
- and is capable of perceiving the state of that environment as a vector of features.
- The machine can execute actions in non-terminal states.
- Different actions bring different rewards and could also move the machine to another state of the environment.
- A common goal of a reinforcement learning algorithm is to learn an optimal policy.

# Model evaluation

# Evaluating Supervised Learning Algorithm

- If you have a set of training data that includes the value you're trying to predict you don't have to guess if the resulting model is good or not.
- If you have enough training data, you can split it into two parts: a training set and a test set.
- You then train the model using only the training set
- And then measure (using r squared or some other metric) the model's accuracy by asking it to predict values for the test set, and compare that to the known, true values.



# Evaluation metrics

- **Accuracy**: the proportion of the total number of correct predictions

$$Accuracy = \frac{Number\ of\ Correct\ predictions}{Total\ number\ of\ predictions\ made}$$

- **Precision and recall** : he proportion of actual positive cases which are correctly identified.
- **Squared error**

$$MeanSquaredError = \frac{1}{N}\sum_{j=1}^{N}(y_j - \hat{y}_j)^2$$

- **Confusion Matrix**

$$Accuracy = \frac{TruePositive + TrueNegative}{TotalSample}$$

- F1 score
- Cost / Utility
- Margin
- Entropy
- K-L divergence

# Classification Metrics

$$\text{accuracy} = \frac{\#\text{ correct predictions}}{\#\text{ test instances}}$$

$$\text{error} = 1 - \text{accuracy} = \frac{\#\text{ incorrect predictions}}{\#\text{ test instances}}$$

# Confusion Matrix

- Given a dataset of P positive instances and N negative instances:

| | | Predicted Class | |
|---|---|---|---|
| | | Yes | No |
| Actual Class | Yes | TP | FN |
| | No | FP | TN |

$$\text{accuracy} = \frac{TP + TN}{P + N}$$

- Imagine using classifier to identify positive cases (i.e., for information retrieval)

$$\text{precision} = \frac{TP}{TP + FP}$$

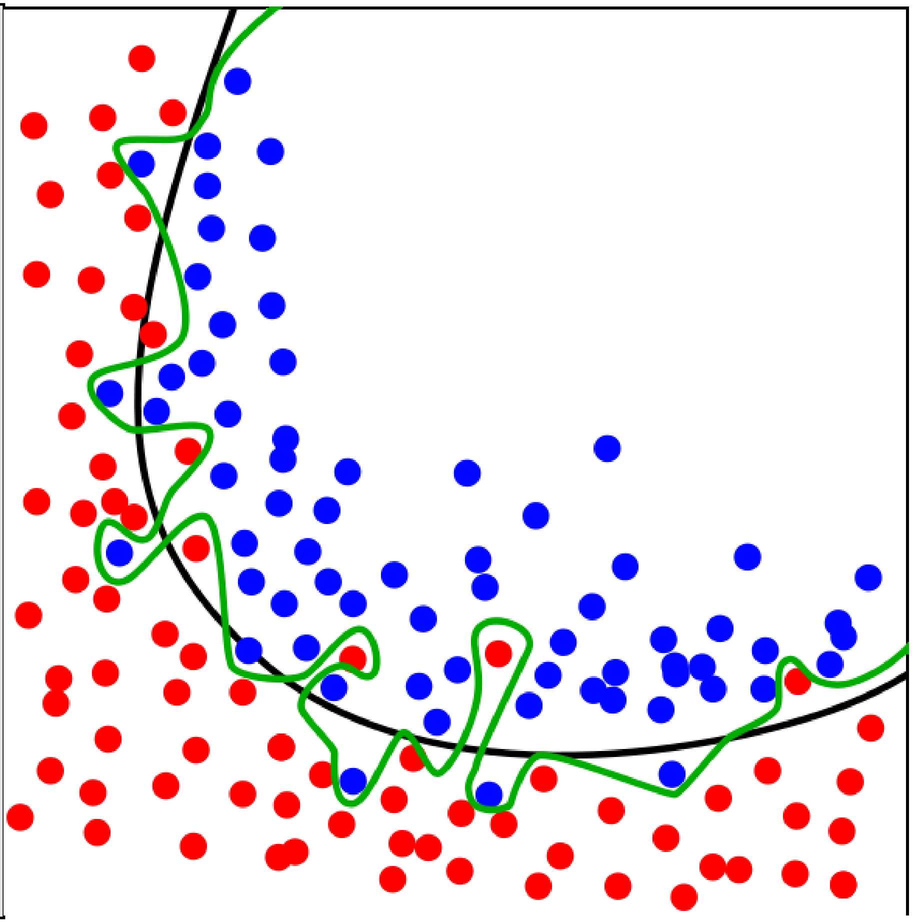
Probability that a randomly selected result is relevant

$$\text{recall} = \frac{TP}{TP + FN}$$

Probability that a randomly selected relevant document is retrieved

# Train/Test in practice

- Need to ensure both sets are large enough to contain representatives of all the variations and outliers in the data you care about
- The data sets must be selected randomly
- Train/test is a great way to guard against overfitting



# K-fold Cross Validation

- One way to further protect against overfitting is K-fold cross validation
- Sounds complicated. But it's a simple idea:
  - Split your data into K randomly assigned segments
  - Reserve one segment as your test data
  - Train on each of the remaining K - 1 segments and measure their performance against the test set
  - Take the average of the K-1 r-squared scores

# When to Use Machine Learning

- When the Problem Is Too Complex for Coding
- When the Problem Is Constantly Changing
- When It Is a Perceptive Problem
- When It Is an Unstudied Phenomenon
- When the Problem Has a Simple Objective
- When It Is Cost-Effective

# When Not to Use Machine Learning

- every action of the system or a decision made by it must be explainable,
- every change in the system’s behavior compared to its past behavior in a similar situation must be explainable,
- the cost of an error made by the system is too high,
- you want to get to the market as fast as possible,
- getting the right data is too hard or impossible,
- you can solve the problem using traditional software development at a lower cost,
- a simple heuristic would work reasonably well,
- the phenomenon has too many outcomes while you cannot get a sufficient amount of examples to represent them (like in video games or word processing software),
- you build a system that will not have to be improved frequently over time,
- you can manually fill an exhaustive lookup table by providing the expected output for any input (that is, the number of possible input values is not too large, or getting outputs is fast and cheap).

# Machine Learning alorithms

- Naïve Bayes Classifier Algorithm (Supervised Learning - Classification)
- K Means Clustering Algorithm (Unsupervised Learning - Clustering)
- Support Vector Machine Algorithm (Supervised Learning - Classification)
- Linear Regression (Supervised Learning/Regression)
- Logistic Regression (Supervised learning – Classification)
- Artificial Neural Networks (Reinforcement Learning)
- Decision Trees (Supervised Learning – Classification/Regression)
- Random Forests (Supervised Learning – Classification/Regression)
- Nearest Neighbours (Supervised Learning)

# Machine Learning Frameworks